

VYSOKÁ ŠKOLÁ BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA  
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Vývoj databázové aplikace pro sportovní obchod

Development of a Database Application for a Sports Shop

Student: Radek Hadaš  
Vedoucí bakalářské práce: RNDr. Antonín Preín

VŠB - Technická univerzita Ostrava  
Ekonomická fakulta  
Katedra aplikované informatiky

## Zadání bakalářské práce

Student: **Radek Hadaš**  
Studijní program: B6209 Systémové inženýrství a informatika  
Studijní obor: 6209R001 Aplikovaná informatika  
Téma: **Vývoj databázové aplikace pro sportovní obchod**  
**Development of a Database Application for a Sports Shop**

Zásady pro vypracování:

1. Úvod
  2. Současné řešení
  3. Realizace projektu
  4. Uživatelský manuál
  5. Závěr
- Seznam použité literatury  
Seznam zkratk  
Prohlášení o využití výsledků bakalářské práce  
Přílohy

Seznam doporučené odborné literatury:


DOBSON, R. *Programování v Microsoft Access 2000: podrobný průvodce programátora tvorbou aplikací v Accessu*. 1. vyd. Praha: Computer Press, 2000. 542 s. ISBN 80-7226-271-8.  
HALVORSON, M. *Visual Basic 6.0 Professional: krok za krokem: naučte se Visual Basic vlastním tempem na reálných příkladech*. 1. vyd. Praha: Computer Press, 1999. 545 s. ISBN 80-7226-169-X.  
KOCICH, P.; SPILKA, O. *1001 tipů a triků pro Microsoft Visual Basic*. 1. vyd. Brno: Computer Press, 2010. 520 s. ISBN 978-80-251-4.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **RNDr. Antonín Prein**

Datum zadání: 26.11.2010

Datum odevzdání: 11.05.2011

  
Ing. Jan Ministr, Ph.D.  
vedoucí katedry



  
prof. Dr. Ing. Dana Dluhošová  
děkanka fakulty

### **Místopřísežné prohlášení**

Místopřísežně prohlašuji, že jsem celou bakalářskou práci vypracoval samostatně. Všechny použité zdroje uvádím v příloze.

Ve Valašském Meziříčí dne.....

Podpis: .....

Radek Hadaš

## **Poděkování**

Děkuji RNDr. Antonínu Preínovi za cenné rady, náměty, připomínky a odbornou pomoc při tvorbě bakalářské práce.

# Obsah

1	Úvod.....	6
2	Současné řešení.....	7
2.1	Komise.....	7
2.2	Prodej.....	8
2.3	Půjčovna .....	10
3	Realizace projektu.....	11
3.1	Microsoft Access 2003 .....	11
3.1.1	Objekty databáze .....	11
3.2	Tabulky databáze .....	14
3.3	Dotaz projektu .....	19
3.4	Visual Basic .....	19
3.2.1	Základní komponenty Visual Basic .....	20
3.5	Realizace pomocí Microsoft Visual Basic.....	20
3.5.1	Komise .....	21
3.5.2	Prodej Zboží .....	28
3.5.3	Půjčovna .....	32
4	Uživatelský manuál.....	37
4.1	Komise.....	38
4.2	Prodej zboží .....	42
4.3	Půjčovna .....	45
5	Závěr .....	48
	Seznam použité literatury .....	49
	Seznam obrázků: .....	50
	Seznam tabulek .....	52
	Seznam použitých zkratk.....	53
	Prohlášení o využití výsledků bakalářské práce.....	54

# 1 Úvod

Koncem loňského roku se mi naskytla příležitost získat brigádu v obchodě, který se přes zimu zabývá prodejem a půjčováním lyžařské výbavy. Protože mě tato nabídka zaujala, lyžování mě baví a zajímalo mě, jak to vypadá na druhé straně barikády, rozhodl jsem se, že nabídku přijmu. Jelikož v té době jsem stál před volbou bakalářské práce a v obchodě jsem se rozkoukal, co a jak funguje, rozhodl jsem se obchodu pomoci i jinak.

Obchod pracuje s velkým množstvím dat, která je potřeba zpravovat. Rozhodl jsem se vytvořit aplikaci, která zjednoduší práci s daty. Proto důležitým rozhodnutím bylo určení nástrojů, v kterých aplikace bude vyvinuta. Bylo potřeba přihlédnout k tomu, že aplikace poběží na obstarožním notebooku. Rozhodl jsem se, že data budou uložena v databázi, která bude vytvořena v MS Access 2003. Pro vývoj samotné aplikace jsem se rozhodl pro starší Microsoft Visual Basic 6.0. Tato aplikace bude využívat data z vytvořené databáze.

V první části bude popsáno současné řešení, jak obchod v současné době pracuje s daty.

Druhá část obsahuje část teoretickou, kdy jsou popsány oba programy. Poté následuje část praktická. Zde jsou popsány jednotlivé tabulky, dotaz a relace databáze. Taktéž jsou popsány všechny formuláře aplikace. Ukázky kódu jsou také přiloženy.

Poslední část obsahuje uživatelský manuál, kde jsou popsány a vysvětleny jednotlivé funkce aplikace.

## 2 Současné řešení

Již od dob založení používá obchod pro ukládání informací o skladových zásobách tabulkovou aplikaci MS Excel. Výběr této aplikace proběhl z důvodu snadné dostupnosti a uživatelské přívětivosti. Dalším důležitým faktorem je také to, že MS Excel je součástí kancelářské sady MS Office.

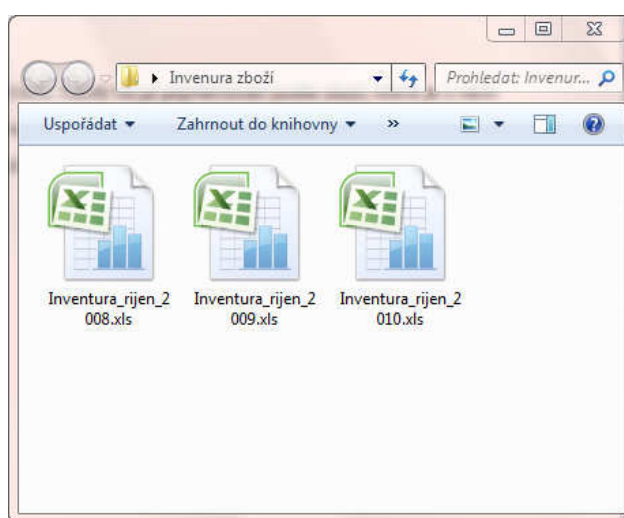
### 2.1 Komise

Obchod neprodává pouze vlastní zboží, ale umožňuje zákazníkům prodat jejich zboží prostřednictvím komise. Jak již bylo zmíněno výše, obchod využívá pro ukládání záznamů o komisi tabulkovou aplikaci MS Excel. Komise pro jednotlivé roky bývá uložena v jednom souboru s názvem “komise\_rokxxx.xls”. Každý tento soubor neboli sešit aplikace MS Excel obsahuje 3 záložky neboli listy.

První list se jmenuje Komisní zboží. Tento list obsahuje tabulku s údaji o daném zákazníkovi a zboží. Záhlaví tabulky obsahuje tyto buňky: datum, jméno a příjmení, adresa, telefon, popis zboží a cenu, za kterou chce zákazník dané zboží prodat. Obsahem tabulky je jednotlivé zboží. Všechny tyto informace se získávají z komisionářské smlouvy, která je vždy se zákazníkem sepsána při převzetí zboží. Dalším listem tohoto souboru je list s názvem prodáno. Tento list obsahuje informace o prodaném komisním zboží. Proto list obsahuje stejnou tabulku, jaká je v listu komisní zboží. Avšak tabulka je doplněna o dva další sloupce. Jedná se o sloupce prodáno dne a prodejní cena. Do této tabulky se zkopírují informace o prodaném zboží z tabulky na listu komisní zboží. Avšak informace se musí doplnit ještě o informace o dni prodeji a prodejní ceně. Posledním listem je list, s názvem vyplaceno. Tento list obsahuje stejnou tabulku jako list s názvem prodáno. Tabulka je ale rozšířena o dva sloupce vyplaceno dne a vyplacena částka. Opět do této tabulky se zkopírují informace ohledně prodaného zboží. Posléze se doplní údaje, kdy bylo zboží vyplaceno a jaká částka byla vyplacena zákazníkovi.

## 2.2 Prodej

Prodej neboli seznam zboží určeného k prodeji je taktéž řešen pomocí aplikace MS Excel. Pro jednotlivé sezóny je vždy k dispozici jednotlivý sešit. Tyto sešity mají názvy podle aktuální sezóny, například *inventura\_rijen\_2010.xls*. Všechny sešity jsou uloženy v jedné složce s názvem *Inventura zboží*. Jednotlivé sešity obsahují určitý počet listů, jejichž množství je odvozeno podle množství druhů zboží, které obchod prodává. Každý list je pro přehlednost zobrazen v jiné barvě.



Obr. 2.1 Složka inventura zboží

Zdroj: vlastní zpracování

Například *inventura\_rijen\_2010.xls* obsahuje 15 listů. Každý list je pojmenován podle zboží, které je v něm zachyceno. Pro představu list číslo jedna se jmenuje *lyže nové*. Tento i další listy obsahují informace o jednotlivých položkách zboží. Tyto informace jsou v každém listu uvedeny v tabulce. Každá tabulka má převážně 11 sloupců. Jednotlivé sloupce obsahují následující informace: inventurní číslo, značka, typ, délka (popřípadě velikost), datum nákupu, nákupní cena s DPH, nákupní cena bez DPH, DPH, daňový doklad, datum prodeje zboží a prodejní cenu zboží. Tyto informace obsahuje list *Nové lyže*, avšak u jiných listů se tyto informace mohou mírně odlišovat v závislosti na typ zboží.



	A	B	C	D	E	F	G	H	I	J	K	L
	Inventurní číslo	Značka	Typ	Délka	Datum nákupu	Cena s DPH	Cena bez DPH	DPH	Daňový doklad	Datum prodeje	Prodejní cena	
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												

Obr. 2.2 Sešit inventura\_rjen\_2010xls

Zdroj: vlastní zpracování

Po přijetí zboží od dodavatele do prodejny se z faktury, popřípadě jiného dokladu vloží do tabulky informace o daném zboží. Konkrétně se jedná o tyto informace: značka, typ, délka, datum nákupu, nákupní cena s DPH, nákupní cena bez DPH, DPH a daňový doklad. Inventurní číslo si vloží obchod dle vlastní režie. Položky datum prodeje a prodejní cena se vloží k danému prodanému zboží až v době prodeje zboží, kdy jsou informace získány z prodejního dokladu. Poté se řádek s prodaným zbožím zvýrazní pro přehlednost oranžovou barvou.

## 2.3 Půjčovna

Půjčovna je další součástí obchodu. Obchod umožňuje zákazníkům si vypůjčit kompletní výbavu na lyže i snowboard. Konkrétně se jedná o tyto části výbavy: lyže, lyžačky, hole, přilba, brýle, snowboard a snowboardové boty. Po vybrání zákazníkem požadované výbavy se všechny části výbavy zapíše do rezervačního formuláře, který je vytisknut na papíře. Dále se doplní informace o zákazníkovi, konkrétně jde o jeho jméno a telefon. Na závěr se doplní datum rezervace.

[illegible]

Obr. 2.3 Formulář rezervace lyží a snowboardů

Zdroj: vlastní zpracování

Po vyzvednutí zákazníkem rezervovaného zboží se zákazníkem sepíše smlouva o půjčení. Smlouva obsahuje údaje o zákazníkovi, konkrétně jde o jeho jméno, adresu, telefon a číslo občanského průkazu. Dále smlouva obsahuje údaje o vypůjčeném zboží, informace o ceně, kterou za vypůjčení zákazník zaplatí a výši zálohy. Smlouvu nakonec zákazník stvrdí svým podpisem. Pro přehlednost se informace o věcech, které si zákazník vypůjčil, zapíše do formuláře o vypůjčených věcech.

### **3 Realizace projektu**

Pro realizaci tohoto projektu bylo rozhodnuto využít databázovou aplikaci, která bude sloužit jako zdroj dat a vývojový program pro vytvoření aplikace, která bude spravovat data z databáze. Proto bylo rozhodnuto, aby se pro ukládání dat využila databázová aplikace balíku MS Office 2003 – MS Access 2003. Pro vlastní vývoj programu byla zvolena aplikace Microsoft Visual Basic verze 6.0. Pro tyto aplikace hovoří fakt, že jsou méně náročné na hardware, na kterém projekt bude běžet. Jejich nevýhodou je však tzv. „duševní zaostalost“.

#### **3.1 Microsoft Access 2003**

MS Access 2003 je aplikace pro správu databází. Všechny informace o databázi jsou uloženy v jednom souboru, nejčastěji s příponou mdb. MS Access 2003 využívá tzv. relační databáze. V relační databázi nejsou totiž data uspořádána do jediné tabulky, ale do několika různých tabulek, které jsou propojeny pomocí relací. Relace jsou logické vazby mezi údaji uloženými v jednotlivých tabulkách [1]. Databáze pro své účely využívá několik základních kamenů neboli objektů.

##### **3.1.1 Objekty databáze**

- Tabulky
- Dotazy
- Formuláře
- Sestavy
- Stránky
- Makra
- Moduly

### **3.1.1.1 Tabulky**

Tabulka je základní objekt každé databáze. Obsahuje veškeré data uložené v databázi. Tabulka se v MS Access 2003 skládá z polí a záznamů. Pole, zjednodušeně řečeno sloupec tabulky, obsahuje identifikační informaci. Každé pole má kromě svého názvu také datový typ. Datový typ rozlišuje, jakýže to údaj se vloží do databáze. MS Access 2003 obsahuje tyto datové typy: text, memo, číslo, datum a čas, měnu, automatické číslo, ano/ne, Objekt OLE, hypertextový odkaz a průvodce vyhledáváním. Každá tabulka musí obsahovat primární klíč. Primární klíč ukazuje na jednoznačnou identifikaci záznamu. Záznamy jsou jednotlivá data uložená v tabulce.

#### **Datové typy**

**Text** - Tento datový typ umožňuje zápis textu, popřípadě čísel. Maximální počet znaků je omezen na 255.

**MEMO** – Taktéž umožňuje vložení textu s kombinací čísel. Oproti datovému typu text je ovšem možné vložit až 65 536 znaků.

**Číslo** – Tento datový typ umožňuje vložení číselných znaků určených pro další výpočty. Maximální velikost je omezena na 8 bajtů.

**Datum a čas** – Datový typ určený pro vložení data nebo času.

**Měna** – Datový typ určený pro peněžní hodnoty. Maximální velikost je 8 bajtů.

**Automatické číslo** – Tento datový typ umožňuje vložení čísel jdoucích po sobě v intervalu 1. Maximální velikost je 4 bajty.

**ANO/NE** – Datový typ určený pro pole, kde hodnoty mohou nabývat hodnot ANO/NE. Hodnota NULL není povolena.

**Objekt OLE** – datový typ umožňuje vložit do pole objekty OLE (dokumenty aplikace MS Word, obrázky, atd.). Maximální velikost je 1 gigabajt.

Hypertextový odkaz – Datový typ umožňuje vložení hypertextového odkazu do pole. Maximální velikost je 64 000 znaků.

Průvodce vyhledáváním – Průvodce slouží k vytváření polí, která posléze umožňují vybrat hodnotu z jiné tabulky nebo ze seznamu hodnot pomocí pole se seznamem. Zvolíte-li tuto možnost ze seznamu datových typů, spustí se průvodce, který dané pole nadefinuje podle vašich pokynů. [2]

### **3.1.1.2 Dotazy**

Dotazy umožňují uživatelům lépe pracovat s tabulkami. Umožňují uživatelům efektivněji vybrat data z tabulek. Uživatel má možnost filtrovat, popřípadě nastavit podmínky, za kterých se vyžadovaná data z tabulky vyberou.

### **3.1.1.3 Formuláře**

Formulář je typem databázového objektu, který je používán hlavně k zadávání dat do databáze nebo k jejich zobrazení. Formulář lze používat také jako přepínací panel, který umožňuje otevření dalších formulářů a sestav v databázi, nebo jako vlastní dialogové okno, které umožňuje přijímat data zadávaná uživateli a provádět s těmito daty různé operace [3]. Nejčastěji jsou na formulář vázány tabulky či dotazy.

### **3.1.1.4 Sestavy**

Sestava umožňuje efektivním způsobem prezentovat informace (například vytištěný dokument) ve formátu jen pro čtení. Jiné objekty aplikace Access, jako jsou formuláře a datové listy, je možné vytisknout, ale sestavy umožňují lépe ovládat způsob zobrazení informací a poskytují větší flexibilitu při prezentaci souhrnných informací [4]. Sestavy mohou obsahovat mnoho prvků, jakým je text, obrázek, graf, atd. Vizuální podoba sestavy je jen na uživateli, jak si sestavu vytvoří.

### 3.1.1.5 Stránky

Je to zvláštní typ webové stránky určený k prohlížení dat z databáze MS Access.

### 3.1.1.6 Makra

Pomocí makra si může uživatel vytvořit komponentu, která pak provede příkaz, který dané komponentě přidá uživatel.

## 3.2 Tabulky databáze

Databáze projektu obsahuje celkem 19 tabulek. Tabulky jsou pro přehlednost pojmenovány podle účelu dat. Tabulky pro komisi a prodej zboží nejsou relační. Tabulky s věcmi pro půjčovnu jsou pomocí relace spojeny s tabulkou informace o zákazníkovi. Všechny tabulky byly vytvořeny pomocí návrhového zobrazení.

Tabulka pro komisi má název Komise. Tabulka obsahuje celkově 12 polí, všechny pole mají kromě názvu pole a datového typu původní nastavení. Primární klíč tabulky je nastaven u pole ID. Tabulka není pomocí relace propojena s žádnou jinou tabulkou.

Název pole	Datový typ
<b>ID</b>	Automatické číslo
<b>Datum</b>	Datum a čas
<b>Jmeno</b>	Text
<b>Prijmeni</b>	Text
<b>Telefon</b>	Číslo
<b>Adresa</b>	Text
<b>PopisZbozi</b>	Text
<b>CenaKProdeji</b>	Měna
<b>ProdanoDne</b>	Datum a čas
<b>ProdejniCena</b>	Měna
<b>VyplacenoDne</b>	Datum a čas
<b>VyplacenoCastka</b>	měna

Tab 3.1 Tabulka komise

Zdroj: vlastní tvorba

Pro prodej zboží slouží 10 tabulek. Ke každému druhu zboží je jedna tabulka. Názvy jednotlivých tabulek jsou: ProdejLyzeNove, ProdejLyzePouzite, ProdejLyzakyNove, ProdejLyzakyPouzite, ProdejHole, ProdejSNB, ProdejSNBBoty, ProdejSNBVazani, ProdejPrilby a ProdejBryle. Každá z uvedených tabulek obsahuje 12 polí. U každé tabulky je jako primární klíč nastaveno pole ID. Všechny tabulky mají níže uvedenou strukturu.

Název pole	Datový typ
<b>ID</b>	Automatické číslo
<b>Značka</b>	Text
<b>Typ</b>	Text
<b>Ročník</b>	Text
<b>Délka</b>	Číslo
<b>DatumNakupu</b>	Datum a čas
<b>NakupsDph</b>	Měna
<b>NakupbezDph</b>	Měna
<b>DPH</b>	Měna
<b>DanovyDoklad</b>	Text
<b>Prodano</b>	Datum a čas
<b>ProdejniCena</b>	Měna

Tab 3.2 Tabulka prodej zboží – lyže

Zdroj: vlastní tvorba

Tabulek pro půjčovnu je vytvořeno celkem 8. 7 tabulek obsahuje informace o jednotlivých věcech k zapůjčení. Jde o tabulky PujcovnaLyze, PujcovnaLyzaky, PujcovnaHole, PujcovnaSNB, PujcovnaSNBBoty, PujcovnaPrilba a PujcovnaBryle. Poslední tabulka obsahuje data o zákazníkovi, který si dané zboží vypůjčuje. Tabulky obsahují 3 popřípadě 4 pole. Pouze tabulka PujcovnaZakaznik obsahuje 12 polí. Primární klíč u každé tabulky je nastaven na pole ID. Ostatní hodnoty jsou na původním nastavení.

<b>Název pole</b>	<b>Datový typ</b>
<b>idLyze</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Typ</b>	Text
<b>Delka</b>	číslo

Tab 3.3 Tabulka půjčovna – lyže

Zdroj: vlastní tvorba

<b>Název pole</b>	<b>Datový typ</b>
<b>idLyzaky</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Typ</b>	Text
<b>Velikost</b>	číslo

Tab 3.4 Tabulka půjčovna – lyžáky

Zdroj: vlastní tvorba

<b>Název pole</b>	<b>Datový typ</b>
<b>idHole</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Delka</b>	číslo

Tab 3.5 Tabulka půjčovna – hole

Zdroj: vlastní tvorba

<b>Název pole</b>	<b>Datový typ</b>
<b>idSNB</b>	Automatické číslo
<b>SNB</b>	Text
<b>Typ</b>	Text
<b>Delka</b>	číslo

Tab 3.6 Tabulka půjčovna - SNB

Zdroj: vlastní tvorba



<b>Název pole</b>	<b>Datový typ</b>
<b>idSNBBoty</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Typ</b>	Text
<b>Velikost</b>	číslo

Tab 3.7 Tabulka půjčovna - SNB Boty

Zdroj: vlastní tvorba

<b>Název pole</b>	<b>Datový typ</b>
<b>idPrilba</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Barva</b>	Text
<b>Velikost</b>	číslo

Tab 3.8 Tabulka půjčovna - Přilba

Zdroj: vlastní tvorba

<b>Název pole</b>	<b>Datový typ</b>
<b>idBryle</b>	Automatické číslo
<b>Znacka</b>	Text
<b>Velikost</b>	Text

Tab 3.9 Tabulka půjčovna – Brýle

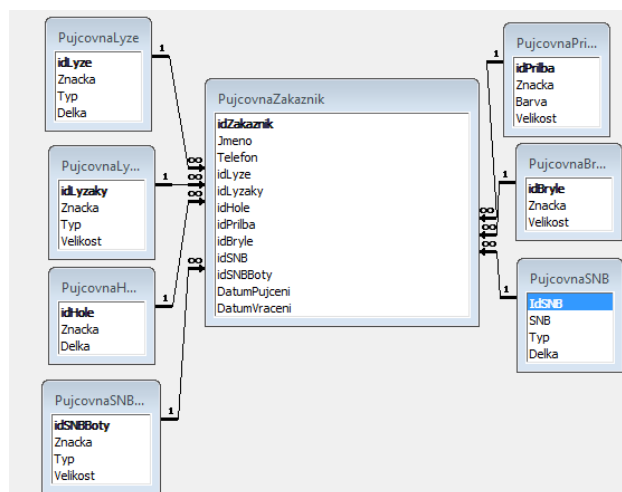
Zdroj: vlastní tvorba

Název pole	Datový typ
<b>idZakaznik</b>	Automatické číslo
<b>Jmeno</b>	Text
<b>Telefon</b>	Číslo
<b>idLyze</b>	Číslo
<b>idLyzaky</b>	Číslo
<b>idHole</b>	Číslo
<b>idPrilba</b>	Číslo
<b>idBryle</b>	Číslo
<b>idSNB</b>	Číslo
<b>idSNBoty</b>	Číslo
<b>DatumPujceni</b>	Číslo
<b>DatumVraceni</b>	Číslo

Tab 3.10 Tabulka půjčovna - Zákazník

Zdroj: vlastní tvorba

Tabulky PujcovnaLyze, PujcovnaLyzaky, PujcovnaHole, PujcovnaSNB, PujcovnaSNBBoty, PujcovnaBryle a PujcovnaPrilba jsou propojeny přes pole id (nastaveno jako primární klíč) s příslušným id polem v tabulce PujcovnaZakaznik. Například pole idBryle v tabulce PujcovnaBryle je propojeno s polem idBryle v tabulce PujcovnaZakaznik. V tomto případě jde o relaci 1:N. V dané relaci byla zajištěna referenční integrita, byla povolena aktualizace souvisejících polí v kaskádě a odstranění souvisejících polí v kaskádě. Typ spojení byl nastaven na hodnotu číslo 2.



Obr. 3.1 Relace

Zdroj: vlastní tvorba

### 3.3 Dotaz projektu

Kromě tabulek bude projekt využívat také jeden dotaz. Dotaz bude sloužit k výběru dat ze všech tabulek, které mají co dočinění s půjčovnou. Z tabulky PujcovnaZakaznik jsou vybrány pole idZakaznika, Jmeno, Telefon, DatumPujceni a DatumVraceni. Z tabulek PujcovnaLyze, PujcovnaLyzaky, PujcovnaHole, PujcovnaSNB, PujcovnaSNBBoty, PujcovnaPrilba a PujcovnaBryle jsou vybrána všechna pole. Z vybraných polí poté nejsou zobrazeny pole s názvem id. Dotaz je uložen pod názvem dotazPujcovna.

### 3.4 Visual Basic

Je událostní programovací jazyk a integrované vývojové prostředí vyvinuté společností Microsoft. Umožňuje vytváření grafických aplikací. Tento programovací jazyk je odvozen od jazyka BASIC. Jeho velkou výhodou je jeho jednoduchost. Zápornou vlastností je ukončení vývoje od Microsoftu, které skončilo v roce 2008. Nástupcem Visual Basic se stal Visual Basic .NET.

Visual Basic je jednoduchý v tom, že v principu jde o skládání komponent na formulář. Těmto komponentám se poté nastavují vlastnosti a akce.

### 3.2.1 Základní komponenty Visual Basic

Label – Jednoduchý popisek, využívá se nejčastěji ve spojení s jinými komponentami, kdy tvoří popis k nim.

Textbox – Textové pole pro vkládání znaků.

CommandButton – Tlačítko.

Checkbox – Zaškrťovací tlačítko, používá se u případů, kdy je potřeba označit více voleb.

OptionButton – Skupina voleb, kdy může být pouze jedna volba označena.

### 3.5 Realizace pomocí Microsoft Visual Basic

Při prvotním spuštění aplikace uživateli nabídne, zda chce vytvořit nový projekt nebo zda chce pokračovat ve vývoji již započatého projektu. Při vytvoření nového projektu New – standart exe se uživateli objeví vývojové prostředí s prázdným formulářem. Prvotní formulář byl pojmenován jako frmSkiSportKasarna. Jako titulek formuláře byl zvolen název Ski Sport Kasárna. Tento formulář bude sloužit jako rozcestník mezi sekcemi aplikace. Proto tento formulář obsahuje 4 tlačítka. První tlačítko přepoše uživatele do sekce komise, druhé tlačítko do sekce prodeje zboží, třetí do půjčovny a poslední tlačítko vypne aplikaci. Tlačítka jsou jednoduše pojmenována podle sekce, do které mají odkazovat. Konkrétně jde o btnKomise, btnProdejZbozi, btnPujcovna a btnKonec. Titulky tlačítek byly přiděleny následujícíce: Komise, Prodej Zboží, Půjčovna a Konec. Každému tlačítku byla přidělena událost, která se vykoná při kliknutí na dané tlačítko. Pro btnKomise je to následující událost po kliknutí: frmKomise.Show. Tato událost bude mít za následek otevření formuláře s názvem frmKomise. U tlačítek btnProdejZbozi je to událost frmProdejZbozi.Show a u tlačítka btnPujcovna je to událost frmPujcovna.Show. Tlačítko konec má velmi jednoduchou událost po kliknutí a to End. Tato událost zapříčiní, že se aplikace vypne. Formulář frmSkiSportKasarna obsahuje také rozbalovací menu, které bylo vytvořeno přes editor menu. Přístup k tomuto editoru je přes Tools – Menu Editor, popřípadě klávesovou zkratkou

CTRL + E. Pomocí tohoto editoru bylo tomuto formuláři přidáno rolovací menu. Menu obsahuje 2 hlavní položky a to Soubor a Nápověda. Položka soubor obsahuje následující podnabídky. Jde o Komise, Prodej Zboží, Půjčovna a Konec. Všechny tyto body podnabídky mají stejnou funkci, jako tlačítka. Projekt byl následně uložen do složky C:\SKISPORT.

### 3.5.1 Komise

Pro komisi byl vytvořen nový formulář. Konkrétně přes Project – Add Form – New – Form. Nový formulář byl pojmenován frmKomise a na tento formulář bude odkazovat tlačítko btnKomise na formuláři frmSkiSportKasarna. Titulek formuláře byl nastaven na Ski Sport Kasárna | Komise. Tento soubor bude sloužit jako zobrazení dat z databáze, konkrétně zobrazení dat z tabulky Komise. Dále zde budou tlačítka, která budou odkazovat na další formuláře určené ke správě komise.

Hlavní komponentou formuláře frmKomise bude zdroj dat, přes který bude probíhat veškerá správa na cestě uživatel – databáze. Tento zdroj není ve standardní nabídce komponentů, proto se musí dodatečně vložit. Komponenta se vkládá přes nabídku Project – Components nebo přes klávesovou zkratku CTRL + T. V nabídce Controls se musí označit volba Microsoft ADO Data Control 6.0 (SP6). Potvrzení klávesou OK v paletě Components přibude v nabídce komponent komponenta Adodc. Vložením této komponenty a následně jejím nakonfigurováním uživatel získá zdroj dat, který může propojit například s databází. Po vložení komponenty Adodc na formulář byla komponenta nakonfigurována. Kliknutím pravým tlačítkem na komponentu a vybráním ADODC Properties se vývojář dostane do možnosti nastavení komponenty. Na stránce General, je třeba vybrat Use Connection String a kliknout na tlačítko Build. Zde se vybere databáze uložena na lokálním disku, v tomto případě v umístění C:\SKISPORT\skisportkasarna.mdb. Tato možnost propojení má nevýhodu v tom, že databáze bude muset být po celou dobu využívání aplikace pořád umístěna v příslušném adresáři. Volba databáze se potvrdí kliknutím na tlačítko OK. Dále v možnostech nastavení komponenty Adodc je třeba se přepnout na záložku Record Source. Na této záložce se musí nastavit Command Type na volbu 2 – adCmdTable a následně vybrat příslušná tabulka z databáze. V tomto případě jde o tabulku Komise. Potvrzením klávesou OK je komponenta Adodc nakonfigurována.

Po nastavení zdroje dat je třeba zvolit, přes jakou komponentu se budou data z tabulky zobrazovat. Volba padla na tabulkové zobrazení dat, podobné jako v Microsoft Access. Komponenta se jmenuje DataGrid. Tato komponenta také není standardní součástí, proto se musí dodatečně přidat. Opět se tak koná pomocí nabídky Project – Components. Oproti předchozí komponentě se označí Microsoft DataGrid Control 6.0 (SP6). Po potvrzení se v komponentech objeví komponenta DataGrid. Po vložení na formulář musí být komponenta nastavena, aby využívala zdroj dat, který byl vložen na formulář. Ve vlastnostech komponenty se vybere možnost DataSource a zde se nastaví název zdroje dat. V tomto případě se vloží název Adodc1. Kliknutím na vloženou komponentu DataGrid1 a kliknutím na ni pravým tlačítkem a vybráním možnosti Retrieve fields se komponenta přizpůsobí tabulce Komise. Poté je třeba ještě komponentu vizuálně upravit. Je záhodno pojmenovat titulky jednotlivých sloupců a popřípadě upravit šířku sloupců tak, aby nebyly příliš široké či úzké. Po spuštění projektu klávesou F5 nebo přes nabídku Run – Start se vývojář může přesvědčit, zda komponenta DataGrid zobrazuje data z tabulky.

Jelikož pouhé zobrazení dat z tabulky by bylo málo, tak je potřeba přidat aplikaci možnosti úpravy dat. V tomto případě jde o tři funkce, přidávání záznamů, editace záznamů a odstranění záznamů. Pro vkládání záznamů do tabulky Komise bude sloužit nový samostatný formulář. Formulář je pojmenován jako frmKomiseNovyZaznam. Přistupovat se k tomuto formuláři dá přes tlačítko btnNovyZaznam, které je umístěno na formuláři frmKomise. Formulář frmKomiseNovyZaznam obsahuje 7 textových polí, 7 popisků k textovým polím a 2 tlačítka. Textové pole budou sloužit pro vložení dat. Textové hodnoty jsou pojmenovány jako txtDatum, txtJmeno, txtPrijmeni, txtTelefon, txtAdresa, txtPopisZbozi a txtCenaKProdeji. U všech těchto polí je nastavena vlastnost Text na prázdnou hodnotu. Popisky k textovým polím jsou pojmenovány následovně: Datum, Jméno, Příjmení, Telefon, Adresa, Popis zboží a Cena k prodeji. Popisky jsou umístěny nalevo od textových polí. Pod popisky a textovými poli jsou umístěny 2 tlačítka. Jde o tlačítka btnOK a btnZpet. Po kliknutí na tlačítko btnZpet se vykoná událost „Me.Hide“, která zavře formulář. Nejprve v editoru kódu je třeba deklarovat tzv. „celoformulářové“ proměnné.

```
Dim conn As ADODB.Connection  
Dim rec As ADODB.Recordset  
Dim esql As String  
Dim esql1 As String
```

Obr. 3.2 Ukázka kódu - celoformulářové proměnné

Zdroj: vlastní zpracování

Dále je potřeba vytvořit pro daný formulář funkci, která se bude starat o připojení k databázi.

```
Private Function connect()  
    Set rec = New ADODB.Recordset  
    Set conn = New ADODB.Connection  
    esql = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & "C:\SKISPORT\SKISPORTKASARNA.mdb" & ";Persist Security Info=False"  
    conn.Open (esql), , , 0  
End Function
```

Obr. 3.3 Ukázka kódu - funkce connect

Zdroj: vlastní zpracování

Poté je třeba do události Form\_Load () zavolat vytvořenou funkci connect. Poté následuje samotná událost, při stisknutí tlačítka btnOK. Nejprve se v události provede kontrola, zda jsou vyplněna všechna textová pole, k tomu slouží následující podmínky.

```
If (txtDatum(0) = "") Then  
    MsgBox "Musíte zadat datum!", vbInformation, "Upozornění"  
ElseIf Not IsDate(txtDatum(0)) Then  
    MsgBox "Zadejte datum ve formátu dd.mm.rrrr!", vbInformation, "Upozornění"  
ElseIf (txtJmeno(1) = "") Then  
    MsgBox "Musíte zadat jméno!", vbInformation, "Upozornění"  
ElseIf (txtPrijmeni(2) = "") Then  
    MsgBox "Musíte zadat příjmení!", vbInformation, "Upozornění"  
ElseIf (txtTelefon(3) = "") Then  
    MsgBox "Musíte zadat telefon!", vbInformation, "Upozornění"  
ElseIf Not IsNumeric(txtTelefon(3)) Then  
    MsgBox "Musíte zadat číslice!", vbInformation, "Upozornění"  
ElseIf (txtAdresa(4) = "") Then  
    MsgBox "Musíte zadat adresu!", vbInformation, "Upozornění"  
ElseIf (txtPopisZbozi(5) = "") Then  
    MsgBox "Musíte zadat popis zboží!", vbInformation, "Upozornění"  
ElseIf (txtCenaKProdeji(6) = "") Then  
    MsgBox "Musíte zadat cenu určenou k prodeji!", vbInformation, "Upozornění"  
ElseIf Not IsNumeric(txtCenaKProdeji(6)) Then  
    MsgBox "Cena musí být číselná!", vbInformation, "Upozornění"
```

Obr. 3.4 Ukázka kódu - podmínky při zadání dat

Zdroj: vlastní zpracování

V těchto podmínkách se zjistí, zda jsou textová pole prázdná. Popřípadě se provede kontrola, zda jsou vložena data dle správného datového typu. Pokud nebyly podmínky splněny, aplikaci vypíše informační dialog s požadovanou chybou. Pokud jsou všechny podmínky splněny, aplikace provede zápis do tabulky Komise. K tomuto účelu slouží metoda Recordset, která je definována v proměnné rec a poté je vytvořena ve funkci connect. Poté co je spuštěna, pomocí vlastnosti .AddNew, provede zápis do jednotlivých polí tabulky. Po samotném zápisu se provede aktualizace tabulky, aktualizace komponenty DataGrid1 na formuláři frmKomise a závěre se formulář.

```

Else
esql1 = "SELECT * from Komise"
rec.Open (esql1), conn, adOpenDynamic, adLockOptimistic
rec.AddNew

    rec.Fields(1) = txtDatum(0)
    rec.Fields(2) = txtJmeno(1)
    rec.Fields(3) = txtPrijmeni(2)
    rec.Fields(4) = txtTelefon(3)
    rec.Fields(5) = txtAdresa(4)
    rec.Fields(6) = txtPopisZbozi(5)
    rec.Fields(7) = txtCenaKProdeji(6)|

rec.Update
If Not rec.EOF Then rec.MoveNext
rec.Close
conn.Close
frmKomise.Adodc1.Refresh
frmKomiseNovyZaznam.Hide
End If

```

Obr. 3.5 Ukázka kódu - zápis do tabulky pomocí Recordsetu

Zdroj: vlastní zpracování

Jelikož se jedná o komisní zboží, které je určeno k dalšímu prodeji, je potřeba do aplikace přidat další modul, který se bude starat o přidání informací o prodeji k již vloženému komisnímu zboží. Tento modul využívá dalšího formuláře, který má název frmKomiseProdatZbozi. Tento jednoduchý formulář obsahuje 2 textová pole, 2 popisky k textovým polím a 2 tlačítka. Textová pole jsou pojmenována txtProdanoDne a txtProdejniCena. Tlačítka mají názvy btnOK a btnZpet. Popisky jsou opět umístěny nalevo od textových polí a nesou hodnoty Prodáno dne a Prodejní cena. Po kliknutí na tlačítko btnZpet se zavolá událost Me.Hide, která zavře dotyčný formulář. Po kliknutí na btnOK se spustí událost, která má na starost samotné vložení hodnot do tabulky. Nejprve se opět musí nadefinovat proměnné, funkce connect a v události Form\_Load() zavolat funkci connect. Teprve poté následuje událost po kliknutí na tlačítko btnOK. Nejdříve se pomocí podmínek zkontroluje, zda jsou náležitě vyplněna textová pole. Pokud jsou podmínky splněny, může se provést samotný zápis do tabulky Komise. Jelikož se nejedná o nový zápis do tabulky, ale aktualizace záznamu stávajícího, je třeba si do proměnné uložit id řádku, do kterého se má zápis provést. Opět se využije vlastnost Recordsetu, tentokrát se jedná o vlastnost .Fields.



```

Dim radek As String
radek = frmKomise.Adodc1.Recordset.Fields("id").Value
|
esql1 = "SELECT * from Komise WHERE ID =" & radek & ""
rec.Open (esql1), conn, adOpenDynamic, adLockOptimistic
rec.Update

    rec.Fields(8) = txtProdanoDne(0)
    rec.Fields(9) = txtProdejniCena(1)

rec.Update
If Not rec.EOF Then rec.MoveNext
rec.Close
conn.Close
frmKomise.Adodc1.Refresh
frmKomiseProdatZbozi.Hide

End If

```

Obr. 3.6 Ukázka kódu - zápis do určitého řádku tabulky

Zdroj: vlastní zpracování

Poté co se vybere řádek podle ID, do kterého se mají oba údaje zapsat, přijde samotná aktualizace záznamu v tabulce. Tentokrát pomocí vlastnosti Recordsetu .Update. Po zapsání do tabulky se aktualizuje tabulka Komise a formulář se zavře.

Pokud je komisní zboží prodáno, může se vlastníkovu komisního zboží vyplatit jeho podíl. K tomuto úkonu slouží další formulář frmKomiseVyplatitZbozi. Formulář se spustí po kliknutí na tlačítko btnVyplatitZbozi na formuláři frmKomise. Tato událost se provede pouze tehdy, když je splněna následující podmínka. Pole ProdanoDne v tabulce Komise musí být u vybraného záznamu vyplněno. Pokud není, vyskočí dialogové okno s informací, že komisní zboží nelze vyplatit. Samotný formulář frmVyplatitZbozi je velmi podobný formuláři frmProdatZbozi. Taktéž obsahuje 2 textové pole, 2 popisky a 2 tlačítka. Textová pole jsou pojmenována txtVyplacenoDne a txtVyplacenoCastka. Popisky jsou umístěny nalevo od textových polí a nesou hodnoty Vyplaceno dne a Vyplacená částka. Pod těmito komponenty jsou umístěna 2 tlačítka btnOK a btnZpet. Tlačítko btnZpet má stejnou funkci, jaká již byla uvedena v předchozích formulářích. Událost při kliknutí na btnOK je stejná, jako událost při kliknutí na btnOK na formuláři frmProdatZbozi. Jediný rozdíl je ten, že se data zapisují do jiných polí. Jde o pole vyplacenoDne a vyplacenoCastka v tabulce Komise.

```

If IsNull(txtProdanoDne(0)) Then
    rec.Fields(8) = ""
Else
    rec.Fields(8) = txtProdanoDne(0)
End If

```

Obr. 3.7 Ukázka kódu - podmínka - prázdný zápis do tabulky

Zdroj: vlastní zpracování

V této chvíli aplikace umožňuje zapisovat data do tabulky Komise. Pokud byla data zadána špatně, je třeba aplikaci umožnit, aby mohla upravovat již zadaná data. Proto byl vytvořen formulář s názvem frmKomiseUpravitZaznam. K tomu formuláři se přistupuje přes kliknutí na tlačítko btnUpravitZaznam, který je umístěn na formuláři frmKomise. Formulář frmKomsieUpravitZaznam obsahuje celkem 11 textových polí, 11 popisků a 2 tlačítka. Textová pole jsou pojmenována txtDatum, txtJmeno, txtPrijmeni, txtTelefon, txtAdresa, txtPopisZbozi, txtCenaKProdeji, txtProdanoDne, txtProdejniCena, txtVyplacenoDne a txtVyplacenoCastka. Popisky jsou umístěny nalevo od textových polí a nesou popisky přidružené vždy k danému textovému poli. Popisky nesou hodnoty datum, jméno, příjmení, telefon, adresa, popis zboží, cena k prodeji, prodáno dne, prodejní cena, vyplaceno dne a vyplacená částka. Tlačítko btnZpet volá událost po kliknutí me.Hide, která zavře formulář. Před událostí, která nastane po kliknutí na tlačítko btnOK je nejprve vytvořen recordset a pomocí něho připojení do tabulky. Událost Form\_Load() kromě spuštění funkce connect obsahuje také vyplnění textových polí daty, která se mají upravit. Také byla přidána podmínka, která zjišťuje, zda bylo již vyplněno textové pole txtProdanoDne. Pokud nebylo, textovému poli txtProdánoDne je přiřazena prázdná hodnota. Poté následuje samotná událost po kliknutí na tlačítko btnOK. Nejprve je nutné zjistit id řádku, který se má upravit. Poté může nastat samotný zápis do příslušného záznamu. Jelikož formulář obsahuje všechna pole tabulky Komise a ne všechny pole záznamu byly před editací vyplněny je třeba pomocí podmínky zobrazené na obrázku zjistit, zda pole bylo prázdné a pokud ano, zapsat do pole prázdný řetězec. Po zápisu do tabulky je nutné tabulku aktualizovat, zavřít recordset a zavřít formulář.

```

txtDatum(0) = frmKomise.Adodc1.Recordset.Fields("datum").Value
txtJmeno(1) = frmKomise.Adodc1.Recordset.Fields("jmeno").Value
txtPrijmeni(2) = frmKomise.Adodc1.Recordset.Fields("prijmeni").Value
txtTelefon(3) = frmKomise.Adodc1.Recordset.Fields("telefon").Value
txtAdresa(4) = frmKomise.Adodc1.Recordset.Fields("adresa").Value
txtPopisZbozi(5) = frmKomise.Adodc1.Recordset.Fields("popisZbozi").Value
txtCenaKProdeji(6) = frmKomise.Adodc1.Recordset.Fields("CenaKProdeji").Value

If IsNull(frmKomise.Adodc1.Recordset.Fields("ProdanoDne").Value) Then
    txtProdanoDne(0) = ""
Else
    txtProdanoDne(0) = frmKomise.Adodc1.Recordset.Fields("ProdanoDne").Value
End If

txtProdejniCena(1) = frmKomise.Adodc1.Recordset.Fields("ProdejniCena").Value

```

Obr. 3.8 Ukázka kódu - zápis do textových polí z polí tabulky

Zdroj: vlastní zpracování

Po přidávání a editaci záznamů je potřeba záznam odstranit. K tomuto účelu slouží událost po kliknutí na tlačítko btnOdstranit na formuláři frmKomise.

```

Private Sub btnOdstranit_Click(Index As Integer)
prompt$ = "Chcete tento záznam odstranit?"
reply = MsgBox(prompt$, vbOKCancel, "Zrušit záznam")
If reply = vbOK Then
    Adodc1.Recordset.Delete
    Adodc1.Recordset.MoveNext
End If
End Sub

```

Obr. 3.9 Ukázka kódu - odstranění záznamu

Zdroj: vlastní zpracování

Po kliknutí na tlačítko btnOdstranit se uživateli zobrazí dialogové okno s informací, zda chce záznam skutečně odstranit. Pokud se rozhodne, že ne, klikne na možnost Storno. Možnost Ok pomocí vlastnosti recordsetu .Delete odstraní příslušný záznam. Poté se ještě pomocí vlastnosti recorsetu .MoveNext přesune na následující záznam.

Kromě výše zmíněných komponent formulář frmKomise obsahuje 2 tlačítka btnZpet a btnKonec. Tlačítko btnZpet volá událost po kliknutí Me.Hide, která zavře formulář. BtnKonec po kliknutí volá událost End, která ukončí aplikaci. Formulář frmKomise také obsahuje rolovací menu, vytvořené pomocí menu editoru. Položky tohoto menu plní stejnou funkci jako tlačítka btnNovyZaznam, btnProdatZbozi, btnVyplatit, btnUpravitZaznam a btnOdstranitZaznam.

### 3.5.2 Prodej Zboží

Sekce prodej zboží slouží k zobrazení kompletního sortimentu zboží, které je určeno k prodeji. K tomuto účelu slouží formulář frmProdejZbozi. Tento formulář se otevře po stisknutí tlačítka btnProdejZbozi na formuláři frmSkiSportKasarna, kdy je zavolána událost frmProdejZbozi.Show.

Formulář frmProdejZbozi obsahuje oproti formuláři frmKomise komponentu SSTab. Tato komponenta funguje na principu záložek. Tato komponenta není standardní součástí, proto je třeba ji přidat přes nabídku Project – Components. Komponenta je součástí balíčku Microsoft Tabbed Dialog Control 6.0. Poté, co byla komponenta přidána na formulář, ji byly upraveny ve vlastnostech komponenty následující vlastnosti. Komponenta byla pojmenována tabProdejZbozi. Počet záložek byl nastaven na číslo 10 podle množství druhů zboží, které obchod prodává. Jednotlivým záložkám byly přidány následující popisky: lyže nové, lyže použité, lyžáky nové, lyžáky použité, SNB, SNB vázání, SNB boty, hole, brýle a přilby. Kromě komponenty SSTab formulář frmProdejZbozi obsahuje celkem 6 tlačítek a rozbalovací menu.

Na každé záložce komponenty SSTab je umístěn zdroj dat, který čte data z tabulek databáze SKISPORTKASARNA.mdb. Každý zdroj dat adodc je propojen s jednou tabulkou databáze. Pro zdroje dat byly zvoleny následující názvy: adoLyzeNove, adoLyzePouzite, adoLyzakyNove, adoLyzakyPouzite, adoSNB, adoSNBVazani, adoSNBBoty, adoHole, adoBryle a adoPrilby. U všech zdrojů dat byla nastavena vlastnost Visible na False. Tato vlastnost zneviditelní danou komponentu na formuláři. Jako zobrazovací komponenta pro zdroj dat je použita komponenta DataGrid. Opět platí pravidlo, jeden zdroj dat, jedna komponenta DataGrid. Jelikož tento formulář má obsahovat celkově desetkrát komponentu DataGrid, je každá pojmenována podle toho, co má zobrazovat. Pro větší přehlednost byla před každý název přidána předpona data. Takže pojmenování je následující: dataLyzeNove, dataLyzePouzite, dataLyzakyNove, dataLyzakyPouzite, dataSNB, dataSNBVazani, dataSNBBoty, dataHole, dataBryle a dataPrilby. Jelikož každá tabulka obsahuje celkem 12 polí a typ jejich obsah je stejný, byla nastavena velikost všech sloupců na stejnou velikost. Pro každý sloupec je nutné ještě nastavit popisek. U všech komponent DataGrid byly nastaveny vlastnosti AllowAddNew, AllowDelete a AllowUpdate na hodnotu False. Tyto vlastnosti

zamezí uživateli editaci buněk. Pro zápis, editaci a zápis do tabulek slouží dodatečné formuláře.

Pro přidávání dat do tabulek slouží formulář frmProdejZboziPridatZaznam. Tento formulář se spustí po zavolání události frmProdejZboziPridatZaznam.Show po kliknutí na tlačítko btnPridatZaznam na formuláři frmKomise. Formulář frmProdejZboziPridatZaznam obsahuje 9 textových polí, 9 popisků a 2 tlačítka. Textová pole mají následující názvy: txtZnacka, txtTyp, txtRocnik, txtDelka, txtDatumNakupu, txtCenaSDPH, txtCenaBezDPH, txtDPH a txtDoklad. Všechny tyto textová pole mají nastavenou vlastnost Text na prázdný řetězec. Popisky jsou jako u předchozích případů umístěny nalevo od textových polí a nesou následující titulky: Značka, Typ, Ročník, Délka, Datum nákupu, Cena s DPH, Cena bez DPH, DPH a Daňový doklad. Formulář doplňují 2 tlačítka s názvy btnOK a btnZpet. Obě tlačítka volají událost po kliknutí.

Jelikož formulář bude zapisovat do tabulky databáze je potřeba vytvořit připojení s databází a Recordset. V události Form\_Load() se kromě samotného spuštění připojení a Recordsetu, upraví u vybraných popisků titulky. Jde o jednoduchou změnu titulku délka na velikost. Poté může následovat samotná událost po kliknutí na tlačítko btnOK. Jelikož všechna textová pole musí být vyplněna, zkontroluje se za pomoci podmínek, zda jsou všechny pole náležitě vyplněna. U textového pole txtDatumNakupu se zkontroluje, zda je zadán údaj ve tvaru data. U textových polí txtCenaSDHP, txtCenaBezDPH a txtDPH se zkontroluje, zda byl zadán číselný údaj. Pokud není ani jedna podmínka splněna, aplikace vypíše pomocí metody MsgBox informační dialog s hlášením, co uživatel zadal špatně. Pokud jsou podmínky splněny, může aplikace začít se zápisem dat do tabulek. Jelikož tento formulář slouží pro zápis do všech 10 tabulek, je potřeba odlišit, do které tabulky se mají data zapsat. O tento problém se starají podmínky.

```

If (frmProdejZbozi.tabProdejZbozi.Tab = 2) Then
    lblDelka(3) = "Velikost:"
ElseIf (frmProdejZbozi.tabProdejZbozi.Tab = 3) Then
    lblDelka(3) = "Velikost:"
ElseIf (frmProdejZbozi.tabProdejZbozi.Tab = 5) Then
    lblDelka(3) = "Velikost:"
ElseIf (frmProdejZbozi.tabProdejZbozi.Tab = 6) Then
    lblDelka(3) = "Velikost:"
ElseIf (frmProdejZbozi.tabProdejZbozi.Tab = 8) Then
    lblDelka(3) = "Velikost:"
ElseIf (frmProdejZbozi.tabProdejZbozi.Tab = 9) Then
    lblDelka(3) = "Velikost:"
End If

```

Obr. 3.10 Ukázka kódu - změna textového pole pomocí podmínek

Zdroj: vlastní zpracování

Podmínka se „zeptá“, zda je vybrána záložka s daným indexovým číslem. U komponenty SSTab jsou záložky číslovány od čísla 0. Pokud je záložka vybrána, může se začít s plněním dat do tabulky. Do proměnné esql1 se pomocí příkazu SELECT vyberou pole, do kterých se má zapisovat. Příkaz SELECT \* FROM ProdejLyzeNove vybere vše z tabulky ProdejLyzeNove. Poté se „otevře“ Recordset, který je definován v proměnné rec. Příkazu se předá příkaz uložený v proměnné esql1. Poté se pomocí vlastnosti Recordsetu .AddNew

```

If (frmProdejZbozi.tabProdejZbozi.Tab = 0) Then

esql1 = "SELECT * from ProdejLyzeNove"
rec.Open (esql1), conn, adOpenDynamic, adLockOptimistic
rec.AddNew

    rec.Fields(1) = txtZnacka(0)
    rec.Fields(2) = txtTyp(1)
    rec.Fields(3) = txtRocnik(2)
    rec.Fields(4) = txtDelka(3)
    rec.Fields(5) = txtDatumNakupu(4)
    rec.Fields(6) = txtCenaSDph(5)
    rec.Fields(7) = txtCenaBezDph(6)
    rec.Fields(8) = txtDPH(7)
    rec.Fields(9) = txtDoklad(8)

rec.Update
If Not rec.EOF Then rec.MoveNext
rec.Close
conn.Close
frmProdejZbozi.adolyzeNove.Refresh
Me.Hide
Unload Me

```

Obr. 3.11 Ukázka kódu - zápis do tabulky ProdejLyzeNove

Zdroj: vlastní zpracování

začne zapisovat do polí, které jsou definovány pomocí vlastnosti Recordsetu .Fields. Do těchto polí se zapíše údaje vložené do textových polí. Po samotném zápisu se uzavře zápis pomocí vlastnosti Recordsetu .Update. Poté se ukončí Recordset a vytvořené připojení do tabulky databáze. Nakonec se zaktualizuje zdroj dat adoLyzeNove. Aktualizací zdroje dat se zaktualizují data v komponentě DataGrid. Nakonec se formulář zavře. Tlačítko btnZpet volá událost po spuštění, kdy se provede příkaz Me.Hide a formulář se zavře.

Další tlačítko, které je umístěno na formuláři frmProdejZbozi je btnProdatZbozi. Po zavolání metody po kliknutí se provede příkaz frmProdejZbozi.ProdatZbozi. Tento formulář obsahuje 2 textové pole, 2 popisky a 2 tlačítka. Textová pole jsou pojmenovány txtProdanoDne a txtProdejniCena. U obou je nastavena vlastnost Text na prázdný řetězec. Popisky jsou umístěny vlevo od textových polí a nesou následující popisky: Prodáno dne a Prodejní cena. Tlačítka jsou opět btnOK a btnZpet. Jelikož se opět jedná o zápis do tabulky, tak je struktura kódu podobná jako v předešlých případech. Tentokrát se akorát nebude jednat o nový záznam, ale o doplnění záznamu stávajícího. Proto je třeba do proměnné radek uložit id řádku, který se má zaktualizovat. Samotný příkaz je zobrazen na obrázku xxx. Dále se pro zápis do tabulky nepoužije vlastnost Recordsetu .AddNew, ale .Update.

```
radek = frmProdejZbozi.adoLyzeNove.Recordset.Fields("id").Value
```

Obr. 3.12 Ukázka kódu - zápis hodnoty "id" do proměnné radek

Zdroj: vlastní zpracování

Pro editaci záznamů v tabulce slouží formulář frmProdejUpravitZaznam. Tento formulář obsahuje 11 textových polí, 11 popisků a 2 tlačítka. Textové pole mají shodné názvy jako na formulářích frmProdejZboziPridatZaznam a frmProdejZboziProdatZbozi. Popisky nesou taktéž shodné titulky jako na již zmíněných formulářích. Tlačítka jsou taktéž totožné. Rozdílů v kódu oproti formuláři frmProdejZboziPridatZaznam není mnoho. Odlišuje se v pouze maličkostech. Prvně jsou pro snadnější editaci záznamů do textových polí načteny hodnoty záznamu, který se má upravit. Tyto hodnoty se do formuláře načtou během zobrazení formuláře, proto je třeba nadefinovat vložení hodnot do události Form\_Load(). Jelikož formulář obsahuje i textová pole txtProdanoDne a txtProdejniCena a uživatel edituje záznam, který tyto pole nemá vyplněna, aplikace nastaví u těchto textových polí vlastnost Enabled na False. Tato vlastnost zamezí jakémukoliv zápisu do těchto polí.

```

If (frmProdejZbozi.tabProdejZbozi.Tab = 0) Then
txtZnacka(0) = frmProdejZbozi.adolyzeNove.Recordset.Fields("Znacka").Value
txtTyp(1) = frmProdejZbozi.adolyzeNove.Recordset.Fields("Typ").Value
txtRocnik(2) = frmProdejZbozi.adolyzeNove.Recordset.Fields("Rocnik").Value
txtDelka(3) = frmProdejZbozi.adolyzeNove.Recordset.Fields("Delka").Value
txtDatumNakupu(4) = frmProdejZbozi.adolyzeNove.Recordset.Fields("DatumNakupu").Value
txtCenaSDph(5) = frmProdejZbozi.adolyzeNove.Recordset.Fields("NakupSDPH").Value
txtCenaBezDph(6) = frmProdejZbozi.adolyzeNove.Recordset.Fields("NakupBezDPH").Value
txtDPH(7) = frmProdejZbozi.adolyzeNove.Recordset.Fields("DPH").Value
txtDoklad(8) = frmProdejZbozi.adolyzeNove.Recordset.Fields("DanovyDoklad").Value
If IsNull(frmProdejZbozi.adolyzeNove.Recordset.Fields("Prodano").Value) Then
txtProdanoDne(0).Enabled = False
Else
txtProdanoDne(0) = frmProdejZbozi.adolyzeNove.Recordset.Fields("Prodano").Value
End If
If (frmProdejZbozi.adolyzeNove.Recordset.Fields("ProdejniCena").Value = 0) Then
txtProdejniCena(0).Enabled = False
Else
txtProdejniCena(0) = frmProdejZbozi.adolyzeNove.Recordset.Fields("ProdejniCena").Value
End If

```

Obr. 3.13 Ukázka kódu - načtení hodnot do textových polí

Zdroj: vlastní zpracování

Tlačítko btnOdtrazitZaznam, jak už název vypovídá, slouží k odstranění vybraného záznamu. Nejprve je ovšem nutno zjistit, na které záložce je daný záznam označen. K tomuto účelu slouží podmínka, která zjišťuje, zda je dotyčná záložka aktivní. Pokud je, záznam se pomocí vlastnosti Recordset .Delete smaže. Poslední dvě tlačítka na formuláři frmProdejZbozi jsou btnZpet a btnKonec. První tlačítko, formulář zavře, druhé ukončí aplikaci. Formulář také obsahuje rolovací menu, které obsahuje nabídku položek, které plní stejnou funkci jako tlačítka na formuláři.

### 3.5.3 Půjčovna

Poslední částí projektu je sekce půjčovna. K tomuto účelu slouží formulář frmPujcovna. Tento formulář taktéž bude zobrazovat data. Tentokrát ovšem ne z tabulky, ale z dotazu dotazPujcovna. Dotaz zobrazuje data z několika tabulek, proto je potřeba vytvořit ještě jeden datový zdroj, který se bude starat o zápis do tabulek. Proto formulář obsahuje dva datové zdroje adoZakaznik a adoZakaznikTabulka. AdoZakaznik je propojen s již zmíněným dotazem, adozakazniktabulka je propojen s tabulkou PujcovnaZakaznik. Dále formulář obsahuje komponentu DataGrid, která je propojena s datovým zdrojem adoZakaznik a zobrazuje data. Formulář dále obsahuje 6 tlačítek, z nichž 2 z nich jsou známé tlačítka btnZpet a btnKonec. Poslední částí formuláře je rolovací menu.



Jelikož si zákazník vybírá věci z půjčovny, tak je třeba přidat možnost přidávat, popřípadě odebírat věci, které obchod půjčuje. Protože jsou věci k půjčení uloženy ve více tabulkách, je součástí aplikace formulář frmPujcovnaOptionTable. Tento formulář slouží jako rozcestník, který uživatele nasměruje k možnosti úpravy seznamu věcí k půjčení. Formulář obsahuje skupinu voleb, kdy uživatel může označit pouze jednu možnost. Těchto možností je celkem 7. Každá možnost má popisek a také název. Výpis možností je následující: lyže (název je opnLyze), lyžáky (opnLyzaky), hole (opnHole), SNB (opnSNB), SNB boty (opnSNBBoty), prilba (opnPrilba) a brýle (opnBryle). Formulář také obsahuje 2 tlačítka. Jedno tlačítko je btnZpet a druhé btnOK. Po vybrání možnosti a stisknutí tlačítka btnOK se provede následující syntaxe příkazů. Za použití podmínek se zjistí, která možnost je aktivní. Podle toho se příkazem nazevformulare.Show aktivuje daný formulář.

```

If opnLyze(0).Value = True Then
frmPujcovnaLyze.Show
ElseIf opnLyzaky(1).Value = True Then
frmPujcovnaLyzaky.Show
ElseIf opnHole(2).Value = True Then
frmPujcovnaHole.Show
ElseIf opnSNB(3).Value = True Then
frmPujcovnaSNB.Show
ElseIf opnSNBBoty(4).Value = True Then
frmPujcovnaSNBBoty.Show
ElseIf opnPrilba(5).Value = True Then
frmPujcovnaPrilba.Show
ElseIf opnBryle(6).Value = True Then
frmPujcovnaBryle.Show
End If

```

Obr. 3.14 Ukázka kódu - skupina voleb

Zdroj: vlastní zpracování

Jelikož možností voleb zobrazení seznamu věcí na půjčení je 7. Bude zde popsána pouze jedna možnost volby. Ostatní možnosti jsou stejné, avšak využívají jiné tabulky databáze. Formulář frmPujcovnaLyze obsahuje zdroj dat adoPujcovnaLyze, komponentu DataGrid s názvem dataPujcovnaLyze a 4 tlačítka, btnPridatZaznam, btnUpravitZaznam, btnOdstranitZaznam, btnZpet a btnKonec. Zdroj dat adoPujcovnaLyze je propojen s tabulkou PujcovnaLyze. Na zdroj dat adoPujcovnaLyze je napojena komponenta dataPujcovnaLyze, která zobrazuje všechny pole z tabulky. Tlačítka btnZpet a btnKonec mají standardní funkci, kdy první zavře formulář a druhé ukončí aplikaci. Událost po kliknutí na tlačítko btnPridatZaznam vykoná příkaz frmPujcovnaLyzePridatZaznam.Show.

Tento formulář obsahuje 3 textové pole, 3 popisky a 2 tlačítka. Textová pole mají názvy txtZnacka, txtTyp a txtDelka. Jejich vlastnost Text obsahuje prázdný řetězec. Popisky jsou na standardním místě nalevo od textových polí. Jejich titulky jsou značka, typ a délka. Tlačítko btnZpet opět zavře formulář. Kliknutím na tlačítko btnOK se provede zápis do tabulky PujcovnaLyze pomocí metody Recordset.

Formulář frmPujcovnaLyzeUpravitZaznam je po vizuální stránce totožný, jako formulář frmPujcovnaPridatZaznam. Po funkční stránce se odlišuje pouze tím, že do textových polí při zobrazení formuláře načte hodnoty záznamu, který se má upravit.

Tlačítko btnOdstranitZaznam na formuláři frmPujcovnaLyze maže aktuální záznam pomocí metody Recordset.

Nový záznam o vypůjčení lyžařského vybavení se vkládá pomocí formuláře frmPujcovnaNovePujceni. Tento formulář obsahuje komponentu ComboBox. Kromě této komponenty, která je na formuláři sedmkrát, formulář obsahuje 4 textové pole, 11 popisků a 2 tlačítka. Názvy textových polí jsou txtJmeno, txtTelefon, txtDatumVypujceni, txtDatumVraceni. Názvy tzv. comboboxů jsou comLyze, comLyzaky, comHole, comSNB, comSNBBotx, comPrilba a comBryle. Tyto komponenty doplňují popisky, které jsou umístěny nalevo od textových polí a comboboxů. Textová pole i comboboxy mají vlastnost Text nastavenou na prázdný řetězec. Comboxy obsahují seznam věcí určených k půjčení. Proto v události Form\_Load() je potřeba tyto comboxy zaplnit. ComboBox se plní vlastností .AddItem. Jelikož počet množství předmětů určených k půjčení je nutné projít jejich seznam cyklem for. Pomocí vlastnosti RecordCount Recordsetu se zjistí počet záznamů v tabulce.

```
Dim i As Integer

For i = 1 To frmPujcovnaLyze.adoPujcovnaLyze.Recordset.RecordCount
    comLyze(0).AddItem Str(frmPujcovnaLyze.adoPujcovnaLyze.Recordset.Fields("idlyze").Value)
    + " | " + (frmPujcovnaLyze.adoPujcovnaLyze.Recordset.Fields("znacka").Value) + " | " +
    frmPujcovnaLyze.adoPujcovnaLyze.Recordset.MoveNext
Next
```

Obr. 3.15 Ukázka kódu - cyklus for

Zdroj: vlastní tvorba

Poté se pomocí cyklu for a vlastnosti comboboxu .AddItem zapíše do jednotlivých comboboxů řádky s údaji tabulky. Tyto údaje se získají ze zdroje dat, umístěného na příslušném formuláři dané věci k půjčení. Pro lepší přehlednost je do řádků zapsán údaj o id věci, značka, typ a délka. Tyto údaje jsou od sebe odděleny operátorem + a vloženými prázdnými znaky a znakem |. Za údajem id je vloženo 5 prázdných znaků, pro pozdější separaci zobrazeného id. Jelikož se bude opět zapisovat do tabulky, je potřeba nejdříve vytvořit spojení s tabulkou PujcovnaZakaznik a vytvořit Recordset. Po spuštění události po kliku na tlačítko btnOK je potřeba nejprve definovat proměnné.

```
Dim lyze As String
Dim lyzaky As String
Dim hole As String
Dim snb As String
Dim snbboty As String
Dim prilba As String
Dim bryle As String
Dim datumvraceni As Date
Dim datumvypujceni As Date
Dim idlyze As Integer
Dim idlyze1 As Integer|
Dim datumvraceni1 As Date
Dim datumvraceni2 As Date
Dim datumpujcen1 As Date
Dim datumpujcen2 As Date
```

Obr. 3.16 Ukázka kódu - definice proměnných

Zdroj: vlastní zpracování

Poté se podmínkami zkontroluje, zda jsou vyplněny textová pole txtJmeno, txtTelefon, txtDatumPujceni a txtDatumVraceni. Pokud nejsou vyplněny nebo pokud jsou vyplněny špatně, aplikace pomocí MsgBoxu informuje uživatele dialogovým hlášením, kde je chyba. Poté, co jsou podmínky splněny, je potřeba získat údaje z comboboxů. Jelikož pro lepší orientaci je v comboboxu zobrazeno více informací a v takovém stavu by nebyl možný zápis do tabulky, je potřeba vyseparovat údaj o id zvolené věci, který se posléze zapíše do tabulky PujcovnaZakaznik. Do proměnných se pomocí vlastnosti Mid\$ vloží příslušné id. První údaj závorce je název comboboxu, druhé číslo je číslo znaku, od kterého se má separovat a poslední číslo značí, kolik znaků se bude separovat. Poté může začít samotný zápis do tabulky PujcovnaZakaznik pomocí Recordsetu. Údaje jméno, telefon, datum půjčení a datum vrácení jsou povinné, avšak zákazník si může vypůjčit pouze jednu věc. Proto je potřeba podmínkou

ošetřit zápis půjčených věcí. Pokud combobox zůstane prázdný, nezapiše se do tabulky nic. Po ukončení zápisu dojde k aktualizaci zdroje dat adoPujcovnaTabulka.

```
lyze = Trim(Mid$(comLyze(0), 1, 4))
lyzaky = Mid$(comLyzaky(1), 1, 4)
hole = Mid$(comHole(2), 1, 4)
snb = Mid$(comSNB(3), 1, 4)
snbboty = Mid$(comSNBBoty(4), 1, 4)
prilba = Mid$(comPrilba(5), 1, 4)
bryle = Mid$(comBryle(6), 1, 4)
```

Obr. 3.17 Ukázka kódu - uložení do proměnných separovaný údaj ID

Zdroj: vlastní zpracování

Spolu s tím se automaticky zaktualizuje dotaz dotazPujcovna. Poté se ukončí Recordset, propojení s tabulkou a formulář se zavře.

Pokud byly údaje o půjčení chybně vloženy, existuje možnost vložená data opravit pomocí formuláře frmPujcovnaUpravitZaznam. Tento formulář je volán příkazem frmPujcovnaUpravitZaznam.Show v události po kliknutí tlačítka btnUpravitZaznam na formuláři frmPujcovna. Formulář je po vizuální stránce stejný jako formulář pro vkládání dat. Po funkční stránce se pouze při otevření formuláře do textových polí a comboboxů vloží údaje z tabulky PujcovnaZakaznik.

Pokud je potřeba záznam odstranit, existuje možnost přes tlačítko btnOdstranitZaznam, kdy po kliknutí vyskočí dialogové okno, zda uživatel chce daný záznam odstranit.

## 4 Uživatelský manuál

Po spuštění spouštěcího souboru xxx.exe, který je umístěn ve složce SKISPORT v kořenovém adresáři disku C: se otevře hlavní okno aplikace. Okno obsahuje tlačítka Komise, Prodej Zboží a Půjčovna. Tlačítka odkazují do příslušných sekcí aplikace. Poslední tlačítko Konec aplikaci ukončí. Přístup do požadovaných sekcí je také umožněn přes menu. Po kliknutí na



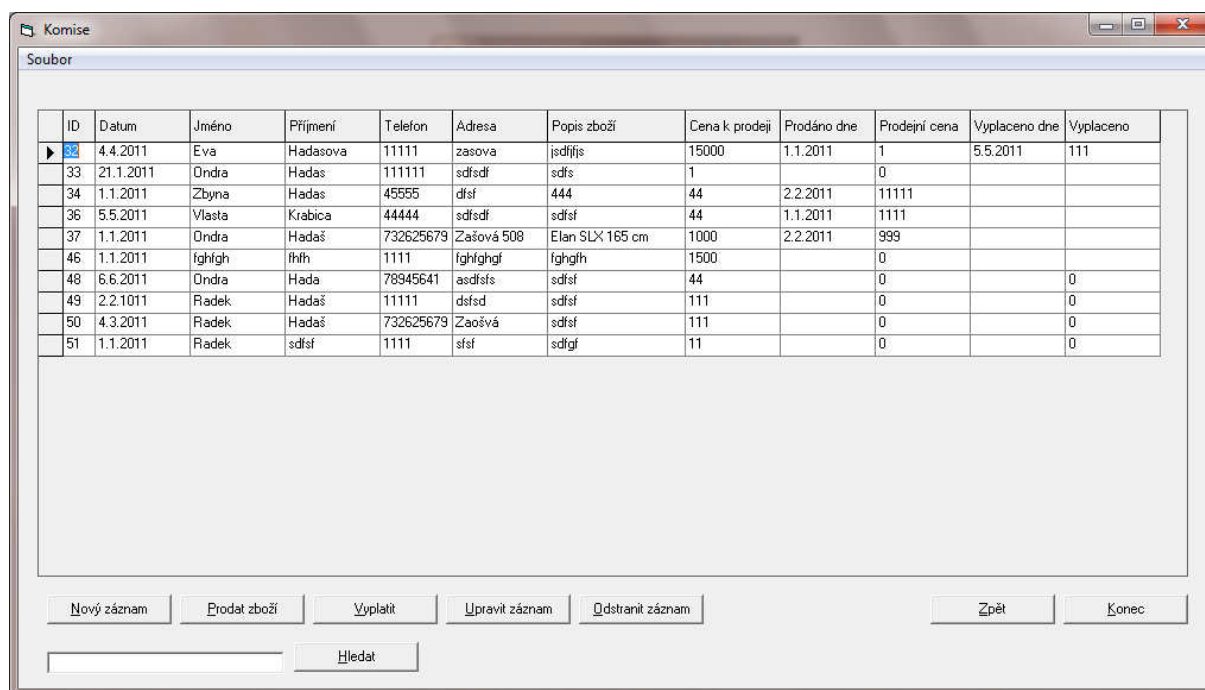
Obr. 4.1 Okno - Ski Sport Kasárna

Zdroj: vlastní tvorba

položku Start se zobrazí možnosti Komise, Prodej zboží, Půjčovna a Konec. Položka menu nápověda obsahuje informaci o aplikaci.

## 4.1 Komise

Po kliknutí na tlačítko Komise se uživateli zobrazující následující pohled. Oknu dominuje uskupení buněk, známé především z aplikace Microsoft Excel. Toto uskupení obsahuje všechny informace o komisi. Pokud uživatel chce nějakým způsobem manipulovat s daty, tak k tomuto účelu slouží tlačítka umístěná pod zobrazením dat.



Obr. 4.2 Okno – Komise

Zdroj: vlastní tvorba

Pokud uživatel chce přidat nový záznam do komise, má 3 možnosti, jak k tomu docílit. První možností je kliknout na tlačítko Nový záznam, druhou možností je použít klávesovou zkratku ALT + N. Poslední možností je přístup přes menu Soubor – Nový záznam. Po spuštění okna nového záznamu je uživatel vyzván k vložení dat. K úspěšnému vložení dat je potřeba vyplnit všechny požadované údaje správně. Pokud uživatel nevyplní příslušné pole, popřípadě ho vyplní špatně, je aplikací vyzván k nápravě.

Obr. 4.3 Okno - Komise nový záznam

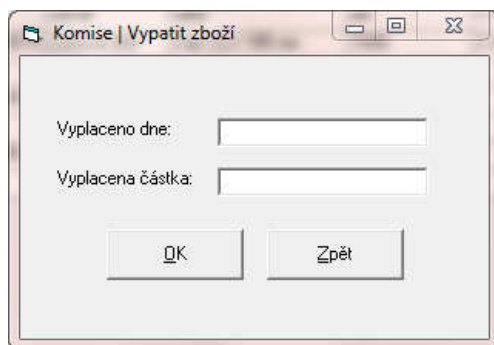
Zdroj: vlastní tvorba

Pokud je komisní zboží prodáno, má uživatel možnost upravit záznam přidáním informací o prodeji. K tomuto účelu slouží okno Prodat zboží. Přístup k tomuto oknu je přes tlačítko Prodat zboží, klávesou zkratku ALT + P nebo přes Soubor – Prodat zboží. Uživatel je poté vyzván k vyplnění informací o dni prodeje a prodejní ceně. Oba údaje musí být vyplněny. Pokud uživatel zadá do pole určeného ke dni prodeji něco jiného, než je datum, je vyzván, aby vložil datum ve správném formátu. Do pole určeného pro prodejní cenu je uživatel povinen vložít pouze číselné znaky.

Obr. 4.4 Okno - Komise prodat zboží

Zdroj: vlastní tvorba

Když je komisní zboží prodáno, je možné vyplatit majiteli jeho podíl. Vyplatit komisní zboží jde přes tlačítko Vyplatit, klávesovou zkratku ALT + T nebo přes Soubor – Vyplatit. Pokud uživateli vyskočí dialogové okno s hlášením, že zboží nelze vyplatit musí nejdříve dané zboží prodat. Pro vyplacení komise je potřeba vyplnit oba údaje o dni vyplacení a o částce, která byla vyplacena. Jestliže uživateli vyskočí chybové hlášení, že zadal některý z údajů špatně, je povinen ho vyplnit správně.



Obr. 4.5 Okno - Komise vyplatit zboží

Zdroj: vlastní tvorba

Jestliže uživatel vyplnil data chybně, popřípadě chce udělat korekci dat, tak má možnost editovat již vložený záznam. K tomuto úkonu se dostane přes tlačítko Upravit záznam, klávesovou zkratku ALT + U, popřípadě přes možnost v menu Soubor – Upravit záznam. Po otevření okna, bude okno automaticky vyplněno již uloženými daty z tabulky. Platí zde stejná pravidla, jako při vkládání nového záznamu. Všechny pole musí být vyplněna, popřípadě musí být data vložena ve správném formátu. Pro uložení hodnot poté uživatel stiskne tlačítko OK, pro volbu neukládat a vrátit se zpět tlačítko Zpět.



Komise | Upravit záznam

Datum:	11.1.2011
Jméno:	Ondra
Příjmení:	Hadaš
Telefon:	732625679
Adresa:	Zašová 508
Popis zboží:	Elan SLX 165 cm
Cena k prodeji:	1000
Prodáno dne:	2.2.2011
Prodejní cena:	999
Vyplaceno dne:	
Vyplaceno:	

OK Zpět

Obr. 4.6 Okno - Komise upravit záznam

Zdroj: vlastní tvorba

Pro odstranění aktuálního záznamu má uživatel možnost kliknout na tlačítko Odstranit záznam. Vyzve ho dialogové okno, zda chce záznam opravdu smazat. Pokud klikne na OK, záznam se smaže, STORNO zruší dialogové okno a vrátí se na hlavní okno komise.

## 4.2 Prodej zboží

ID	Značka	Typ	Ročník	Délka	Datum nákupu	Cena s DPH	Cena bez DPH	DPH	Doklad	Prodáno dne	Prodejní cena
2	Elan	SLX	10/11	165	1.1.2011	18 000,00 Kč	15 000,00 Kč	3 000,00 Kč	1011125	2.2.2012	15 000,00 Kč
5	Elan	WaveFlex14	10/11	176	2.2.2011	15 000,00 Kč	3 000,00 Kč	1 111,00 Kč			0,00 Kč
6	Elan	Race SL	10/11	165	3.3.2011	15 000,00 Kč	555,00 Kč	1 144,00 Kč		2.2.2011	1 000,00 Kč
7	Elan	82xTi	10/11	168	24.12.2011	16 000,00 Kč	12 000,00 Kč	2 000,00 Kč	1151d		0,00 Kč
8	Elan	Black Perla	10/11	152	12.12.2011	5 000,00 Kč	4 000,00 Kč	1 000,00 Kč	250115mnh		0,00 Kč
9	Elan	WaveFlex 8	10/11	160	3.3.2011	15 000,00 Kč	12 000,00 Kč	3 000,00 Kč	111212s		0,00 Kč
10	Elan	WaveMagic	10/11	152	2.2.2011	12 000,00 Kč	1 000,00 Kč	2 000,00 Kč	11111d		0,00 Kč
11	Elan	E-Flex4	10/11	160	1.1.2011	4 000,00 Kč	3 000,00 Kč	1 000,00 Kč	dsfsd010		0,00 Kč
12	Elan	White Magic	10/11	152	12.12.2011	5 000,00 Kč	4 000,00 Kč	1 000,00 Kč	250115mnh		0,00 Kč

Obr. 4.7 Okno - Prodej zboží

Zdroj: vlastní tvorba

Jestli si uživatel bude chtít zobrazit skladové zásoby zboží, které je určené k prodeji, má možnost kliknout v hlavním menu na položku Prodej zboží. Popřípadě použít klávesovou zkratku ALT + P. Sekce prodej zboží je rozdělena přehledně 10 záložkami. Každá záložka obsahuje jednotlivý druh zboží. Vše je přehledně zobrazeno v tabulce. Uživatel má dále možnost přidávat záznamy, upravovat je, popřípadě záznam odstranit. Pro přidání záznamu stačí kliknout na tlačítko Přidat záznam, použít klávesovou zkratku ALT + P, popřípadě použít rozbalovací menu

Prodej zboží | Přidat záznam

Značka:

Typ:

Ročník:

Délka:

Datum nákupu:

Cena s DPH:

Cena bez DPH:

DPH:

Daňový doklad:

OK Zpět

Obr. 4.8 Okno - Prodej zboží - přidat záznam

Zdroj: vlastní tvorba

Pro přidání zboží do skladu je potřeba správně vyplnit všechny položky v okně. Pokud se tak nestane, uživatel je upozorněn dialogovým oknem. Přidání uživatel potvrdí tlačítkem OK. Záznam je také možné zpětně opravit. Stačí využít k tomu tlačítko Upravit záznam, po kliknutí se otevře editační okno. Zde platí stejná pravidla, jako pro přidávání. Při otevření jsou načtena do textových polí data, která se mají editovat.

Prodej | Upravit záznam

Značka:

Typ:

Ročník:

Délka:

Datum nákupu:

Cena s DPH:

Cena bez DPH:

DPH:

Daňový doklad:

Prodáno dne:

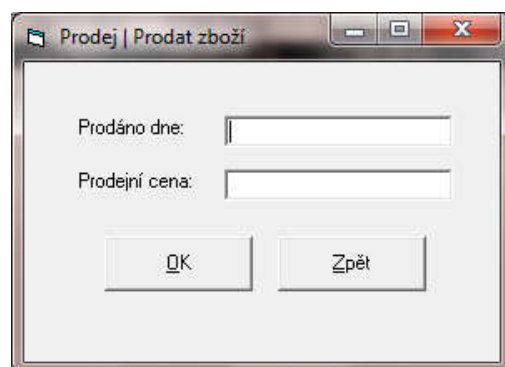
Prodejní cena:

OK Zpět

Obr. 4.9 Okno - Prodej zboží - upravit záznam

Zdroj: vlastní tvorba

Záznam je také možné odstranit. Stačí jednoduše kliknout na tlačítko Odstranit záznam a potvrdit OK. Když se zboží prodá, má uživatel možnost k již zadanému zboží přidat informaci o data prodeji a částku, za jakou se zboží prodalo. Musí ovšem vyplnit správně údaje, pokud se tak nestane, je informován.

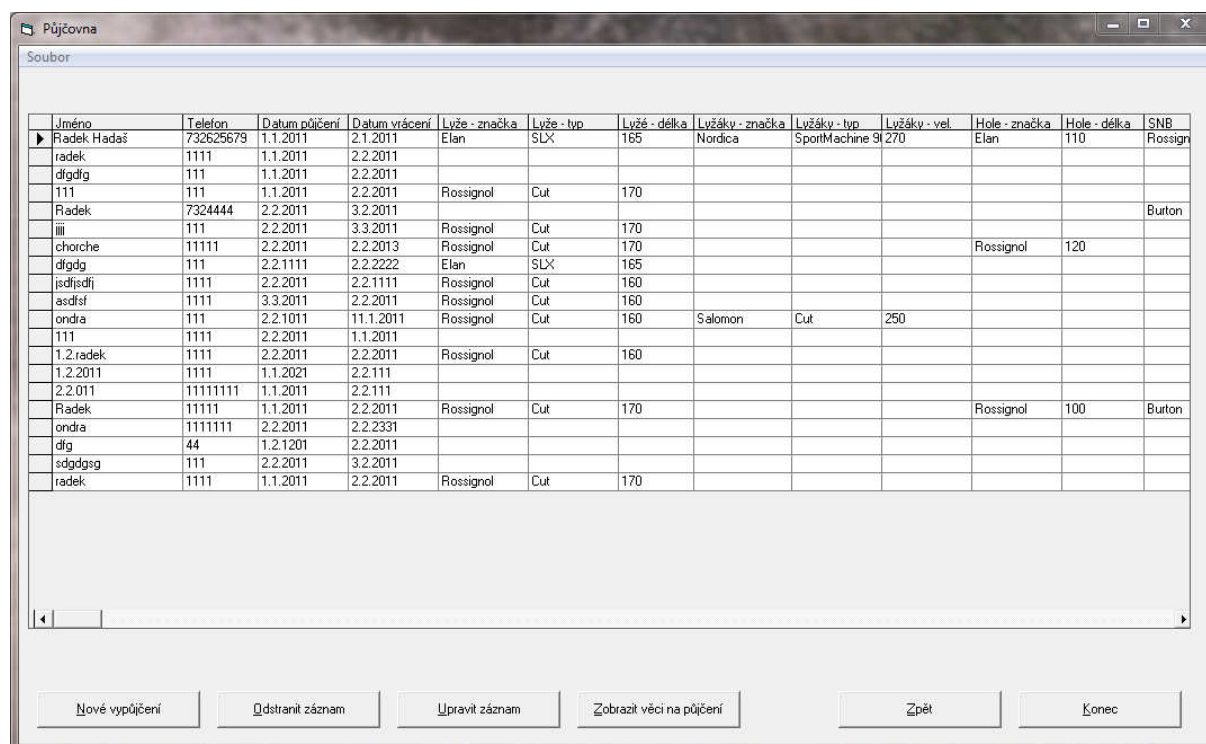
The image shows a standard Windows-style dialog box. The title bar at the top reads 'Prodej | Prodat zboží' and includes standard minimize, maximize, and close buttons. The main area of the dialog is light gray and contains two labels with corresponding text input fields: 'Prodáno dne:' followed by an empty text box, and 'Prodejní cena:' followed by another empty text box. At the bottom of the dialog, there are two buttons: 'OK' on the left and 'Zpět' (Back) on the right. Both buttons have a light gray background and a thin border.

Obr. 4.10 Okno - Prodej zboží - prodat zboží

Zdroj: vlastní zdroj

### 4.3 Půjčovna

Pro snadnou evidenci půjčeného zboží slouží uživateli sekce půjčovna. Jsou zde přehledně zobrazeny informace o půjčujícím, jeho telefon, datum vypůjčení a datum vrácení. Samozřejmostí je kompletní zobrazení všech položek, které si dotyčný vypůjčil. Uživatel má také možnost editovat seznamy věcí, které jsou k půjčení.



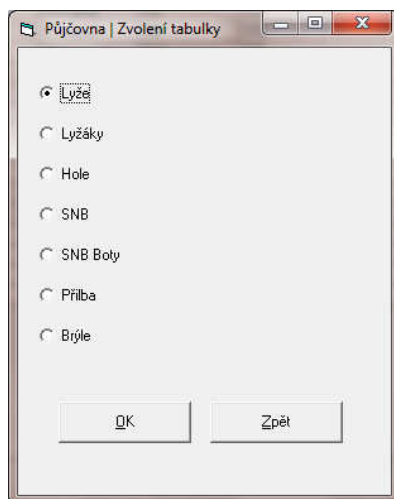
The screenshot shows a window titled 'Půjčovna' with a menu bar containing 'Soubor'. Below the menu bar is a table with 13 columns: Jméno, Telefon, Datum půjčení, Datum vrácení, Lyže - značka, Lyže - typ, Lyže - délka, Lyžáky - značka, Lyžáky - typ, Lyžáky - vel., Hole - značka, Hole - délka, and SNB. The table contains 20 rows of data. At the bottom of the window, there are six buttons: 'Nové vypůjčení', 'Odstranit záznam', 'Upravit záznam', 'Zobrazit věci na půjčení', 'Zpět', and 'Konec'.

Jméno	Telefon	Datum půjčení	Datum vrácení	Lyže - značka	Lyže - typ	Lyže - délka	Lyžáky - značka	Lyžáky - typ	Lyžáky - vel.	Hole - značka	Hole - délka	SNB
► Radek Hadaš	732626679	1.1.2011	2.1.2011	Elan	SLX	165	Nordica	SportMachine 91	270	Elan	110	Rossignol
radek	1111	1.1.2011	2.2.2011									
dfgdfg	111	1.1.2011	2.2.2011									
111	111	1.1.2011	2.2.2011	Rossignol	Cut	170						
Radek	7324444	2.2.2011	3.2.2011									Burton
iii	111	2.2.2011	3.3.2011	Rossignol	Cut	170						
chorche	11111	2.2.2011	2.2.2013	Rossignol	Cut	170				Rossignol	120	
dfgdfg	111	2.2.1111	2.2.222	Elan	SLX	165						
jsdfjsdfj	1111	2.2.2011	2.2.1111	Rossignol	Cut	160						
asdfsfs	1111	3.3.2011	2.2.2011	Rossignol	Cut	160						
ondra	111	2.2.1011	11.1.2011	Rossignol	Cut	160	Salomon	Cut	250			
111	1111	2.2.2011	1.1.2011									
1.2.radek	1111	2.2.2011	2.2.2011	Rossignol	Cut	160						
1.2.2011	1111	1.1.2021	2.2.111									
2.2.011	11111111	1.1.2011	2.2.111									
Radek	11111	1.1.2011	2.2.2011	Rossignol	Cut	170				Rossignol	100	Burton
ondra	11111111	2.2.2011	2.2.2331									
dfg	44	1.2.1201	2.2.2011									
sdgdfgsg	111	2.2.2011	3.2.2011									
radek	1111	1.1.2011	2.2.2011	Rossignol	Cut	170						

Obr. 4.11 Okno – Půjčovna

Zdroj: vlastní tvorba

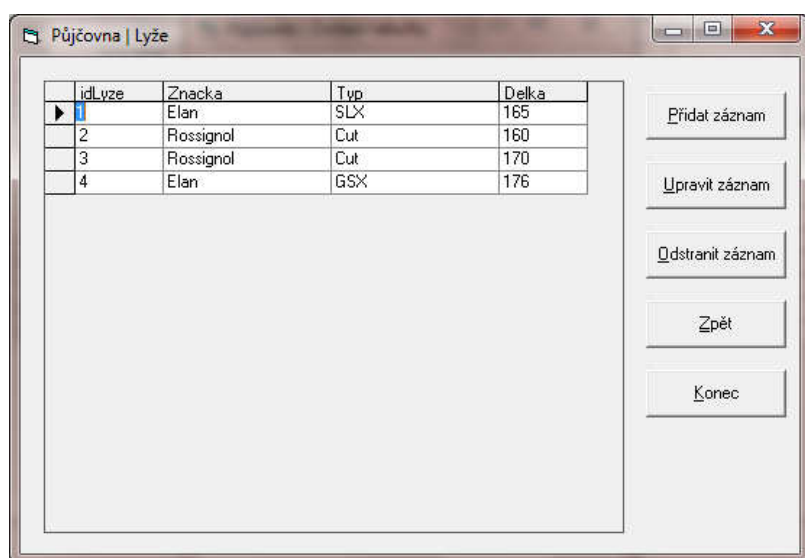
Po kliknutí na tlačítko Zobrazit věci na půjčení má uživatel možnost vybrat druh výbavy na půjčení, které chce editovat. Jednoduchým kliknutím na požadovaný druh a následně na tlačítko OK, se uživatel dostane do možnosti editace.



Obr. 4.12 Okno - Půjčovna - zvolení tabulky

Zdroj: vlastní zpracování

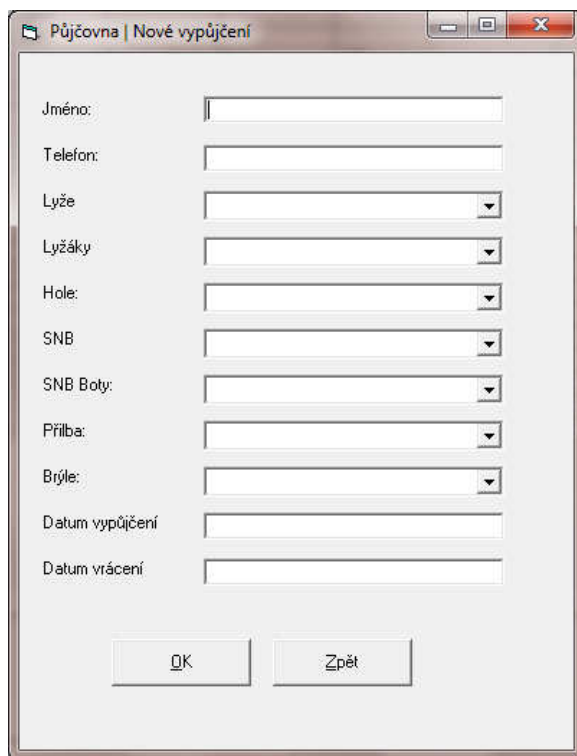
Po zvolení předmětu editace se uživateli zobrazí okno s kompletním seznamem výbavy k zapůjčení. Dále má uživatel možnost přidat do tohoto seznamu další položky, upravit jejich vlastnosti, popřípadě položku odstranit. Položku přidá kliknutím na tlačítko Přidat záznam. Otevře se nové okno, kde uživatel musí vyplnit údaje o značce, typu, délce, popřípadě velikosti. Dále má uživatel možnost položku upravit. Jednoduchý kliknutím na tlačítko Upravit záznam se otevře stejné okno jako u tlačítka Přidat záznam. Rozdíl je pouze v tom, že při otevření okna se zobrazí data o editovaném předmětu. Položku odstraní kliknutím na tlačítko Odstranit záznam. Tento postup funguje u všech druhů věcí k zapůjčení.



Obr. 4.13 Okno - Půjčovna – lyže

Zdroj: vlastní tvorba

Pro přidání informací o zákazníkovi, který si výbavu vypůjčí, slouží formulář Nové vypůjčení, na který se uživatel dostane kliknutím na tlačítko Nové vypůjčení. Na tomto formuláři uživatel povinně vyplní jméno, telefon, datum vypůjčení a datum vrácení. Pak je jen na zákazníkovi, jaké součásti výbavy si vybere.



The image shows a software window titled "Půjčovna | Nové vypůjčení". It contains a form with the following fields and controls:

- Jméno: [Text input field]
- Telefon: [Text input field]
- Lyže: [Dropdown menu]
- Lyžáky: [Dropdown menu]
- Hole: [Dropdown menu]
- SNB: [Dropdown menu]
- SNB Boty: [Dropdown menu]
- Přilba: [Dropdown menu]
- Brýle: [Dropdown menu]
- Datum vypůjčení: [Text input field]
- Datum vrácení: [Text input field]
- Buttons: "OK" and "Zpět" (Back)

Obr. 4.14 Okno - Půjčovna - nové vypůjčení

Zdroj: vlastní tvorba

## 5 Závěr

Cílem práce bylo vytvoření aplikace, která obchodu pomůže se záznamem dat o komisi, prodeji zboží a půjčovny. Pomocí aplikací MS Access 2003 MS Visual Basic 6.0 se podařilo vytvořit aplikaci, která využívá a spravuje data z databáze. Oproti stávajícímu řešení, jsou všechna data uložena na jednom místě. Taktéž možnost práce se záznamy je situována do jedné aplikace.

Záznamy o komisi jsou přehledně zobrazeny v tabulce a pomocí dodatečných formulářů může uživatel záznamy přidávat, editovat a mazat. Vše je řešeno naprosto intuitivně a „user friendly“. Pokud uživatel zadá nějaký záznam chybně, uživatel je informován, kde že to udělal chybu.

Seznam zboží pro prodej je řešen podobně, jako komise. Jelikož druhů zboží k prodeji je více, jsou tabulky zobrazeny pomocí záložek. Taktéž zde má uživatel možnost přidávat, editovat a mazat záznamy.

V půjčovně má uživatel editovat seznam výbavy, která je zákazníkům k dispozici. Pomocí formulářů se přidávají, editují a mažou záznamy o vypůjčených věcech. Seznam vypůjčených věcí je přehledně zobrazen v tabulce, která je na hlavním formuláři půjčovny.

Aplikace splňuje všechny požadavky na správu všech potřebných dat. Pokud bude chtít obchod aplikaci upravit, popřípadě přidat další vylepšení, je to možné. Jelikož všechny požadavky na aplikaci byly splněny, je možné dát aplikaci do užívání.



## Seznam použité literatury

### Monografické zdroje:

[1] PÍSEK, Slavoj. *Access 2003: jak poznáme databáze, jak vytváříme tabulky, jak pracujeme s formuláři, jak tiskneme sestavy, jak používáme filtry, jak tvoříme dotazy*. 1.vyd. Praha: Grada, 2004. 124s. ISBN 80-247-0787-X

### Elektronické zdroje:

[2] Datové typy polí v aplikaci Microsoft Access (MDB) [online]. 2007. [cit. 2011-04-10]. Dostupný z WWW: <<http://office.microsoft.com/cs-cz/access-help/datove-typy-poli-v-aplikaci-microsoft-access-mdb-HP005238518.aspx>>

[3] Formuláře [online]. 2007. [cit. 2011-04-12]. Dostupný z WWW: <<http://office.microsoft.com/cs-cz/access-help/formulare-HP005275182.aspx>>

[4] Sestava – základní pojmy [online]. 2007. [cit. 2011-04-12]. Dostupný z WWW: <<http://office.microsoft.com/cs-cz/access-help/sestava-zakladni-pojmy-HA001154964.aspx>>

## Seznam obrázků:

Obr. 2.1 Složka inventura zboží .....	8
Obr. 2.2 Sešit inventura_rijen_2010xls .....	9
Obr. 2.3 Formulář rezervace lyží a snowboardů .....	10
Obr. 3.1 Relace .....	19
Obr. 3.2 Ukázka kódu - celoformulářové proměnné .....	22
Obr. 3.3 Ukázka kódu - funkce connect .....	23
Obr. 3.4 Ukázka kódu - podmínky při zadání dat .....	23
Obr. 3.5 Ukázka kódu - zápis do tabulky pomocí Recordsetu .....	24
Obr. 3.6 Ukázka kódu - zápis do určitého řádku tabulky .....	25
Obr. 3.7 Ukázka kódu - podmínka - prázdný zápis do tabulky .....	26
Obr. 3.8 Ukázka kódu - zápis do textových polí z polí tabulky .....	27
Obr. 3.9 Ukázka kódu - odstranění záznamu .....	27
Obr. 3.10 Ukázka kódu - změna textového pole pomocí podmínek .....	30
Obr. 3.11 Ukázka kódu - zápis do tabulky ProdejLyzeNove .....	30
Obr. 3.12 Ukázka kódu - zápis hodnoty "id" do proměnné radek .....	31
Obr. 3.13 Ukázka kódu - načtení hodnot do textových polí .....	32
Obr. 3.14 Ukázka kódu - skupina voleb .....	33
Obr. 3.15 Ukázka kódu - cyklus for .....	34
Obr. 3.16 Ukázka kódu - definice proměnných .....	35
Obr. 3.17 Ukázka kódu - uložení do proměnných separovaný údaj ID .....	36
Obr. 4.1 Okno - Ski Sport Kasárna .....	37
Obr. 4.2 Okno – Komise .....	38
Obr. 4.3 Okno - Komise nový záznam .....	39
Obr. 4.4 Okno - Komise prodat zboží .....	39
Obr. 4.5 Okno - Komise vyplatit zboží .....	40
Obr. 4.6 Okno - Komise upravit záznam .....	41
Obr. 4.7 Okno - Prodej zboží .....	42
Obr. 4.8 Okno - Prodej zboží - přidat záznam .....	43
Obr. 4.9 Okno - Prodej zboží - upravit záznam .....	43
Obr. 4.10 Okno - Prodej zboží - prodat zboží .....	44
Obr. 4.11 Okno – Půjčovna .....	45

Obr. 4.12 Okno - Půjčovna - zvolení tabulky .....	46
Obr. 4.13 Okno - Půjčovna – lyže .....	46
Obr. 4.14 Okno - Půjčovna - nové vypůjčení.....	47

## Seznam tabulek

Tab 3.1 Tabulka komise .....	14
Tab 3.2 Tabulka prodej zboží – lyže .....	15
Tab 3.3 Tabulka půjčovna – lyže .....	16
Tab 3.4 Tabulka půjčovna – lyžáky .....	16
Tab 3.5 Tabulka půjčovna – hole .....	16
Tab 3.6 Tabulka půjčovna - SNB .....	16
Tab 3.7 Tabulka půjčovna - SNB Boty .....	17
Tab 3.8 Tabulka půjčovna - Přilba .....	17
Tab 3.9 Tabulka půjčovna – Brýle .....	17
Tab 3.10 Tabulka půjčovna - Zákazník.....	18

## **Seznam použitých zkratek**

DPH    Daň z přidané hodnoty

SNB    Snowboard

MS    Microsoft

## Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne .....

.....

jméno a příjmení studenta

Adresa trvalého pobytu studenta:

.....



